IE 506: Machine Learning - Principles and Techniques **Jan-May 2024**

Project End-term Report : Knowledge Transfer via Multiple Model Local Structure Mapping

Team Name: Mayur_B *Team Members: Mayur R Bhurle (23M1027)*

Abstract

The project is about implementing and experimenting with LOCALLY WEIGHTED ENSEMBLE (LWE) ALGORITHM, there are many Ensemble Algorithms such as Bagging, Boosting etc. but they are Globally Weighted Ensembles, which give equal weight to all the base models. In the Locally Weighted Ensemble algorithm we give different weights to different models, based on their ability to predict the data points belonging to different classes. In Mid-Term Review the LWE algorithm has been implemented on the 20 Newsgroup dataset in order to test the performance of the model and compare its performance with the individual base models, SMA (Simple Model Averaging) Framework and LS-SVM (Least Square Support Vector Machine) model. Further as a part of the End-term review I have implemented the Locally Weighted Ensemble algorithm on stock market dataset that has temporal dependencies in order to check if the Locally Weighted Ensemble Algorithm is applicable on the dataset that has inter-dependencies. It was found that the LWE algorithm performed poorly on the dataset, however the proposed Confidence based Ensemble Algorithm outperformed all other methods.

1 Introduction

The researchers have developed Locally Weighted Ensemble (LWE) algorithm which is a novel algorithm that combines multiple models to make predictions. It is useful in transfer learning problems where training and test domains are different. The LWE assigns weights to individual models based on their local behaviors at each test example, which allows it to adapt to different test examples.

The motivation behind this project is to improve the prediction accuracy in situations where the training and test domains differ, a common scenario in many real-world applications. This report provides an overview of the project, detailing the different steps in the algorithm, the experiments conducted, and the results.

The report is structured as follows: Section 2 surveys the relevant literature focusing on Sample Selection Bias and Covariate Shift , Section 3 describes the algorithm, Section describes the work done Section 4 provides details on the dataset used, Section 5 outlines the experiment performed, Section 6 presents the results, section 7 provides the work done after the Mid-term Review and Section 8 concludes the report.

2 Literature Survey

Traditional ensemble methods assign model weights based on the training set or use fixed prior weights. However, these methods may not be suitable for transfer learning problems where the training and test domains follow different distributions. The following authors discuss the same issue.

Fan et al. (2006) [1]: This paper explores how data selection bias can affect building accurate models. In knowledge transfer, we are using a model trained on one dataset (source domain) to predict on another dataset (target domain). If the data collection processes for these datasets are different, it can lead to selection bias. The paper proposes using model averaging and unlabeled data from the target domain to reduce this bias and improve the transferred knowledge.

Huang et al. (2007) [2]: Similar to Fan et al., this paper also focuses on sample selection bias and how it impacts learning algorithms. In knowledge transfer, this bias can occur if the training data doesn't well-represent the data we want to predict on. The paper introduces a method for correcting this bias by matching the distributions of the source and target data in a special mathematical space. This can improve the transfer of knowledge by reducing the mismatch between the training and testing data.

Shimodaira (2000) [3]: This paper deals with a specific type of covariate shift where the factors influencing the data (covariates) differ between the source and target domains. In knowledge transfer, this can happen if the contexts or situations where the data was collected are very different. The paper proposes a technique for weighting the training data to account for this difference and improve the model's generalizability to the target domain.

Basically Fan et al. [1] and Huang et al. [2] talk about Sample Selection Bias which means that the training data doesn't represent the full range of possibilities you might encounter later. And Shimodaira [3] talks about Covariate Shift which means that the conditions or context where the data was collected are different between the training and testing situations.

3 Methods and Approaches

3.1 Clustering Manifold Assumption:

The researchers assume that the dataset follows the clustering manifold assumption according to which for a sample x if the neighbouring points belong to class y then the sample x also belongs to class y.

The LWE uses a graph-based approach to approximate the optimal per example weight under the “clustering-manifold” assumption that P(x) is related to P(y|x). It constructs two graphs for each test example and a base model to be combined, and approximates the model weight as the similarity between the local structures around the test example in the two graphs. The LWE also includes a local structure-based adjustment mechanism to handle cases where the concepts carried by all the models conflict with the actual concept at the test example, meaning the average similarity value is lesser than a given threshold value.

This section discusses the Locally Weighted Ensemble (LWE) framework, a technique for combining multiple models in situations where the training and testing data come from different distributions (known as transfer learning). Here we'll focus on two key aspects of LWE: graph-based weight estimation and local structure-based adjustment. These steps play a crucial role in assigning appropriate weights to different models and potentially adapting predictions based on local information.

3.2 Graph-Based Weight Estimation (representing partial LWE (pLWE) algorithm):

Let us discuss in steps how Graph based weight estimation method representing the partial Locally weighted algorithm (pLWE algorithm) is implemented,

a. Clustering the Test Set: We start by grouping similar data points in the new dataset into clusters. This helps us understand the underlying structure of the data. The researchers have used a software called CLUTO for clustering, however I have simply used K means Clustering.

b. Building Neighborhood Graphs: In graph based weight estimation, we construct two graphs GT based on clusters made by the clustering algorithm, and GM based on the corresponding models predictions for each base model for all the test samples. Then we compare the similarity between the two graphs. The similarity is normalized in order to calculate the weights that will be assigned to each base model.

c. Weight Calculation: We calculate a similarity score (s(GM, GT; x)) that reflects how well the connections between models (GM) and clusters (GT) match around x. This score essentially tells us how much the model's predictions agree with the local structure of the data. This score is calculated by calculating the number of common points that have been used to make pairs between the two graphs and then dividing this number by the total number of test samples.

d. Assigning Weights: Finally, each model receives a weight $w_{\{Mi,x\}}$ proportional to its similarity score for a specific data point (x). Models with higher scores (better alignment with local structure) get higher weights, signifying their greater influence on the final prediction for x. The mathematical formula given below represents the normalized weight for the base model *i* , for a total number of datapoints *k.*

$$\mathrm{w}_{\{\mathrm{M,x}\}} \propto \mathrm{s}\big(\mathrm{G}_{\{\mathrm{M}\}}, \mathrm{G}_{\{\mathrm{T}\}};\ \mathrm{x}\big)$$

$$w_{\{Mi,x\}} = \frac{s\big(G_{\{Mi\}}, G_{\{T\}};\ x\big)}{\sum_{\{i=1\}}^{\{k\}} s\big(G_{\{M,i\}}, G_{\{T\}};\ x\big)}$$

e. This approach gives more weight to models that consistently predict the same class for data points residing in similar regions(clusters) of the new dataset. This helps LWE leverage the strengths of each model in areas where their predictions are likelyto be accurate. Finally, the prediction for the partial LWE (pLWE) algorithm are made using the formula given below.

$$P(y|E,x) = \sum_{\{i=1\}}^{\{k\}} \big(w_{\{M_i,x\}}\big)\, P(y|M_i,x)$$

3.3 Local Structure-Based Adjustment: When Models Disagree:

While weight estimation helps identify reliable models, there might still be situations where all models perform poorly for a specific data point (x). This could happen if the local structure (cluster) around x doesn't reflect the true class distribution. Here's how LWE handles such scenarios:

1. Average Similarity Score: LWE calculates an average similarity score ($s_{avg(x)}$) for all models around x. This score represents the overall agreement between model predictions and the local structure.
2. Threshold Check: If the average score ($s_{avg(x)}$) falls below a predefined threshold (δ), it suggests that none of the models are reliable for predicting the class of x. This might indicate a region where both models struggle due to limitations in their training data.
3. Leveraging Local Structure: In such cases, LWE discards the model predictions and relies solely on the local structure(cluster) of x in the new dataset. It predicts the class of x by looking at the majority class label of other data points within the same cluster (C) whose predictions from the weighted ensemble (E) were reliable (i.e., their ($s_{avg(x)}$) was above the threshold). The equation given below represents the mathematical formula used to calculate the final probability for the data points for which the similarity value is lesser than the threshold value, where C' represents the cluster which contain the datapoints that have ($s_{avg(x)}$) value greater than the threshold value, U represents the unsupervised classifier (K mean clustering) and $\frac{c(y, C'|E)}{|C'|}$ represents the count of datapoints that have been classified as belonging to class y by the ensemble E, divided by the total number of datapoints in the cluster C'.

$$P(y|U, x\ \in C) \approx \{\frac{P(y, x\ \in C'|E)}{\{P(x\ \in C')\}}\} \approx \{\frac{c(y, C'|E)}{|C'|}\}$$

Finally, argmax function is applied in order to estimate the final prediction. Summarizing this section it can be said that, Local structure-based adjustment acts as a safety net when models disagree, and the local structure suggests a potential mismatch between training and testing data.

3.4 Work done in Mid-Term Review:

In Mid-Term Review I had implemented LWE algorithm on 20 newsgroup dataset in 2 steps that is

Step1: Implementing pLWE (Partial Locally Weighted Ensemble), which is based only on Graph based weight estimation and does not consider Local Structure based adjustment. Graph based weight estimation has been implemented as explained in section 3.2 except for one difference that instead of using CLUTO for clustering (because of unavailability of code), spectral clustering has been used as it also satisfies the Clustering Manifold Assumption (see section 3.1).

Step2. Implementing Local Structure based adjustment (as explained in section 3.3), in this step we first set a threshold which we have set at 0.5, meaning at least 50% of the edges between the two graphs Gm and Gt should be similar. Then we calculate the average similarity for each test sample that is s_avg. So if average similarity for a test sample is greater than 0.5 then the final prediction gotten from pLWE will be used for that test sample if not then we discard the prediction made by the pLWE model and consider a new unsupervised clustering model, (which in our case is K means Clustering) to make the predictions.

K means clustering has been used because the research paper mentions an unsupervised algorithm for clustering that is based on neighbors of x, although KNN would be a more appropriate choice for this but since it does not form clusters I decided to go with K means clustering algorithm. Now all the test samples that have s_avg less than 0.5, are clustered using K means clustering and separated into 4 different test clusters. 4 test clusters because there are 4 sub categories in the test data. Additionally, other iteration have been tried using number of clusters equal to 2 and 6 respectively.

Now after test clusters are formed reliable clusters are made that contain test samples for which the s_avg is greater than 0.35. Here I have taken 0.35 because all test samples have s_avg less than 0.4. So in order to make reliable predictions 0.35 was selected as an appropriate value.

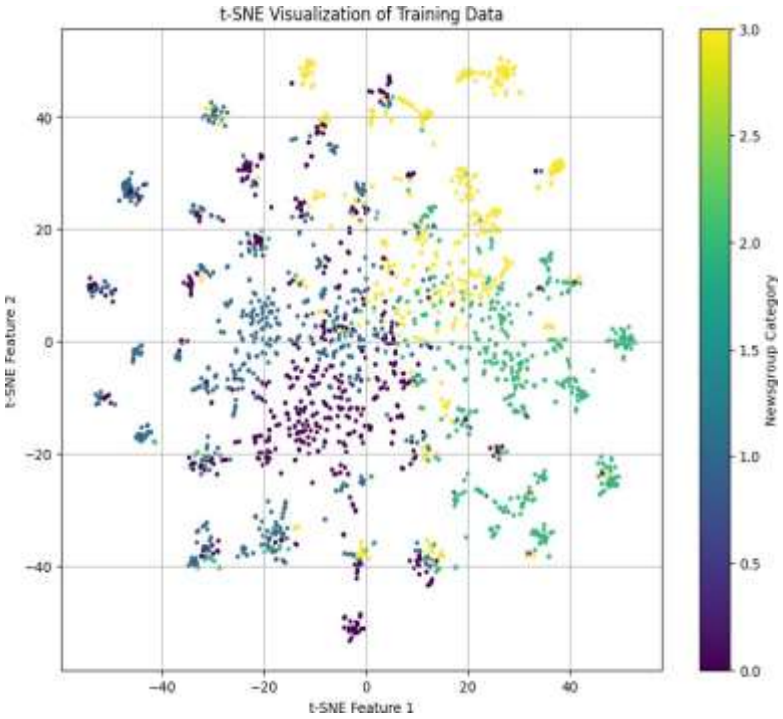
Now we calculate the final probability for a test sample x belonging to say class 0, by dividing the no of reliable samples that also have class label 0, in that cluster to which x belongs, divided by the the total number of reliable test samples in that particular reliable cluster. All the reliable points are put in a cluster called C', called reliable cluster. For each cluster there is a corresponding reliable cluster. For example let us say test example x belongs to a cluster, then probability that x belongs to class 0 = ("no of reliable samples in cluster C' with label 0" / "total number of reliable samples in C' cluster "). At the end, argmax function was applied on the obtained probabilities to get the final predictions.

4 Data set Details

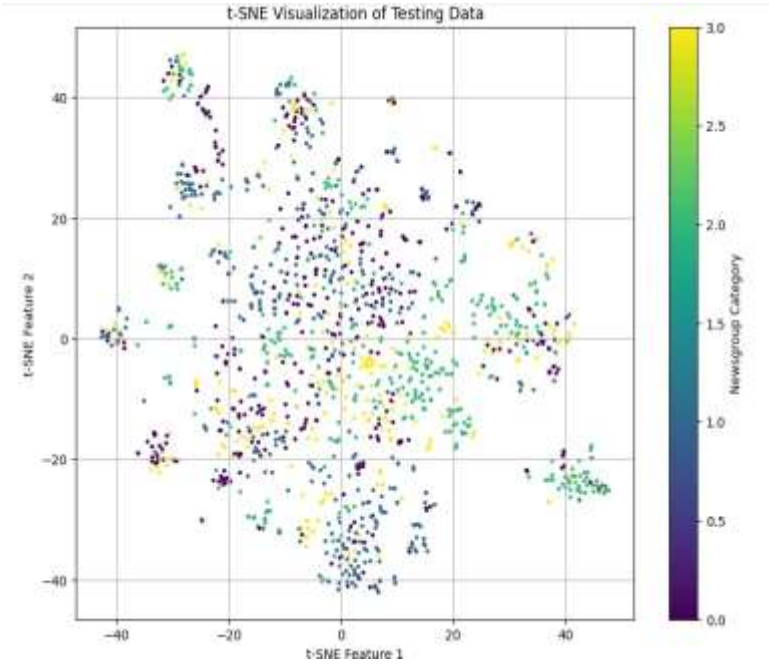
The LWE algorithm is implemented on 20 newsgroup dataset for binary classification. There are 20 different sub categories in the dataset, which are used to make two different classes each consisting of 4 subcategories.

The dataset is textual in nature and for pre-processing the tex is first converted into vectors using the TF-IDF Vectorizer followed by dimensionality reduction using PCA method so that data can be processed faster. After which t-SNE is applied to the PCA-reduced data for both training and testing sets. (t-SNE is a tool to visualize high-dimensional data. t-SNE has a cost function that is not convex, i.e. with different initializations we can get different results).

Figure 1 represents how the subcategories in the 20 newsgroup dataset are spread in a given space. It is clear that the training dataset can be easily clustered compared to the testing dataset. The four different colors represent four different subcategories.



a) Visualizing Training data



b) Visualizing testing data

Figure 1: The two graphs represent the spread of the 4 different subcategories from Rec and Talk categories, for training and testing data.

The 20 newsgroup dataset was imported from scikit using the command “from sklearn.datasets import fetch_20newsgroups”. As mentioned earlier, this dataset is a collection of approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups. The TRAINING AND TESTING dataset had different subcategories an example of which is given below for “Rec vs Talk” classification task.

training_set = ['talk.politics.guns', 'talk.politics.misc', 'rec.autos', 'rec.motorcycles']

testing_set = ['rec.sport.baseball', 'rec.sport.hockey', 'talk.politics.mideast', 'talk.religion.misc']

5 Experiments

The experiment aims to assess the performance of the LWE algorithm against the performance of individual base models and the performance of other algorithms such as SMA, pLWE and LS-SVM.

Decision Tree Classifier, Logistic Regression and Support Vector Machine were used as base models for the LWE algorithm and their performance is also compared with that of the LWE MODEL. The Locally Weighted Ensemble (LWE) framework is compared with several other learning algorithms which are used to baseline the performance, the algorithms are described below:

a) Simple Model Averaging (SMA): This is a basic ensemble method where all models are combined with equal weights.

b) Partial Locally Weighted Ensemble (pLWE): This method uses the weighted ensemble approach from LWE but skips the local structure-based adjustment step.

c) Least Square Support Vector Machines (LS-SVM): This is a supervised learning algorithm which is being used to compare the performance of LWE. The researchers used Transductive Support Vector Machines (T-SVM), however due to unavailability of code LS-SVM was used. The code for LS-SVM was taken from git-hub[7].

6 Results

Algorithm	**Accuracy**					
	C vs S classification	**R vs T classification**	**R vs S classification**	**S vs T classification**	**C vs R classification**	**C vs T classification**
Decision Tree	0.589	0.640	0.651	0.616	0.706	0.725
Logistic Reg.	0.673	0.623	0.728	0.720	0.829	0.892
SVM	0.736	0.659	0.736	0.731	0.848	0.921
SMA	0.694	0.6338	0.7332	0.7156	0.833	0.8931
LS-SVM	0.7232	0.2895	0.7780	0.7803	0.821	0.8828
p-LWE	0.6381	0.6296	0.6967	0.7078	0.7996	0.8541
LWE	0.5973	0.7563	0.4823	0.4542	0.8268	0.8342

Fig 2a. Accuracy obtained for 20 newsgroup dataset on 6 different classification experiments

Algorithm	Mean Squared Error					
	C vs S classification	**R vs T classification**	**R vs S classification**	**S vs T classification**	**C vs R classification**	**C vs T classification**
Decision Tree	0.411	0.360	0.349	0.384	0.294	0.275
Logistic Reg.	0.327	0.377	0.272	0.280	0.171	0.108
SVM	0.264	0.341	0.264	0.269	0.152	0.079
SMA	0.305	0.3661	0.2667	0.2843	0.166	0.1068
LS-SVM	0.2767	0.7104	0.2219	0.2196	0.178	0.1171
p-LWE	0.3618	0.3703	0.3032	0.2921	0.2003	0.1458
LWE	0.4026	0.2536	0.5176	0.2810	0.1731	0.1657

Fig 2b. MSE obtained for 20 newsgroup dataset on 6 different classification experiments

Number of Clusters	Accuracy					
	C vs S classification	**R vs T classification**	**R vs S classification**	**S vs T classification**	**C vs R classification**	**C vs T classification**
2	0.5973	0.7463	0.4823	0.4464	0.8268	0.5383
4	0.5973	0.5242	0.4823	0.4464	0.4987	0.8342
6	0.5973	0.7406	0.4823	0.4470	0.6668	0.5383

Fig 2c. Obtained accuracy for different number of clusters for 6 different classification experiments.

The results shown in figure 2a and 2b represent that the LWE algorithm performed relatively well on the R vs T, C vs R, C vs T classification tasks giving higher accuracy and lower MSE compared to other algorithms. Even though the accuracy and MSE are not as high as compared to the expected results from the research paper which are around 98%, the obtained results prove the point that the LWE algorithm has better performance compared to most of the other algorithms mentioned in section 5.

Number of clusters is a hyperparameter used in the experiments performed. The researchers fixed average similarity value as 0.7 and varied the number of clusters in order to check the variation in performance of the LWE algorithm. It was observed that highest accuracy is obtained for number of cluster equal to 2. I have tried similar variation in the hyperparameters while keeping the average similarity fixed at 0.35(as the average similarity in my case was significantly lower). Figure 2c represents the results obtained from the variations. Unlike the results obtained by the researchers there was no clear pattern observed from the variation in the hyperparameter.

Following are few observations and facts that explain the variations in results:

1. For C vs S, R vs S and S vs T classification tasks the LWE accuracy and MSE is not as high as compared to the expected results which is around 98%. It was noticed that the dataset belonging to S category is

common in all three cases, and the lower performance in the results could be attributed to the discrepancy in the data belonging to the S category, however I could not find any concrete proof for this claim.

2. Another reason could be because of the small changes I have made in the algorithm, such as in the second step of the LWE algorithm, I have used a threshold of 0.35 as the average similarity value in my case was very low, compared to that obtained by the researchers which was around 0.7.
3. Also as mentioned earlier the researcher have used a clustering package called CLUTO, whereas I have used K meansclustering because of the unavailability of CLUTO software, which could also lead to differences in the results.

7 Work Done After Mid-term Review

7.1 Data set:

After Mid-Term based on the reviews obtained, I had to try the LWE algorithm on a dataset that has interdependencies. In order to do that I have selected the stock market data that consists of stocks of multiple companies belonging to different sectors such as Finance, Information Technology (IT) and Energy. Stock market data was chosen as per the suggestion of the respected professor given during the mid term review. Stock market data is time series data and it has temporal dependencies, hence by introducing features that capture the temporal dependencies such as rolling mean, rolling max/min and lag, and then testing the LWE algorithm on this dataset it could be tested if LWE algorithm is suitable for data containing inter-dependencies.

7.2: Experiment

The stocks of different companies were categorized into different classes as described in section 7.1. For example Wipro and Infosys were put in IT class and stocks belonging to companies such as HDFC and AXIS were put in Finance class. This data was used to train the model for binary classification problem in which the model had to predict which stock belonged to Finance class and which belonged to IT. For testing dataset, a completely unseen stock data was given to the model. The data of Bajaj Finance company and TCS company were combined to form the testing data and the algorithm was asked to predict which data points belonged to Finance company stock (Bajaj Finance), and which datapoints belonged to IT company stock (TCS).

7.3 Modifications

In addition to testing the LWE algorithm on dataset with inter-dependencies I also have also made changes in the weighting scheme and developed a new algorithm called Confidence based Ensemble algorithm. Initially in LWE algorithm the weightage is given on the basis of similarity (see section 3.2), however I have given the weightage to the models on the basis of probability given to each datapoint by the model that has correctly classified the datapoint. The weighting scheme is described below. (In the given algorithm X_test represents the test data).

Algorithm: Confidence based Ensemble

Input: 1. Predictions made by Base Models.

2. Probabilities given by Base Models

Output: 1. Weight for each model

2. Final predictions for all data points

Algorithm: FOR each x € X_test,

1. Select the base model with correct predictions

2. Among the models with correct prediction give 100% weightage to the model with high accuracy.

3. If none of the base models give correct prediction then, give 100% weightage to the model with the highest accuracy.

4. Store the weights for base models corresponding each data point.

7.4 Results

Sr. No.	*Algorithm*	*Accuracy*		
		IT vs Finance	**IT vs Energy**	**Finance vs Energy**
1	**LSTM**	0.6154	.3290	0.4453
2	**SVM**	0.3088	0.1491	0.2918
3	**LR**	0.4761	0.1834	0.2763
4	**CNN**	0.3731	0.2625	0.4605
6	**pLWE**	0.4748	0.1327	0.5
7	**LWE**	0.4590	0.3331	0.5
8	**Confidence Based Ensemble**	0.7698	0.4052	0.6102

Fig. 3a Accuracy obtained using Confidence Based Ensemble Algorithm and comparison with other baseline algorithms.

Binary Classification tasks were performed as described in section 7.2 for three different cases that is IT vs Finance, IT vs Energy and Finance vs Energy. It was observed that the proposed Confidence Based Ensemble algorithm gave highest accuracy in all 3 cases compared to 7 different algorithms as shown in fig. 3a. Further the confidence value was changed while combining the predictions of Confidence Based Ensemble algorithm with the predictions given by K means clustering in the Local Structure Based Adjustment step and it was observed that as we increase the confidence level the accuracy in all 3 cases decreases, suggesting that the predictions made by K means clustering are not aiding in the improvement of accuracy. Fig. 3b shows the variation accuracy with confidence.

Confidence(p)	*Accuracy*		
	IT vs Finance	**IT vs Energy**	**Finance vs Energy**
p>0.4	0.7698	0.4052	0.6102
p>0.5	0.7698	0.4052	0.6102
p>0.6	0.6581	0.3735	0.5526
p>0.7	0.4754	0.3670	0.5191
p>0.8	0.4665	0.3649	0.5024
p>0.9	0.4629	0.3632	0.4991

Fig. 3b Variation in Accuracy with confidence

8 Conclusion

- LWE offers an advantage by assigning weights based on local behaviour (using clusters), it leverages the strengths of each model instead of blindly combining predictions that might be inaccurate in certain regions. Furthermore, the local structure-based adjustment provides a backup mechanism when models fail to generalize well using the Local Structure based adjustment step.
- The LWE algorithm is effective for transfer learning problems especially when the training data and testing data are not necessarily similar. It effectively combines multiple models by assigning weights based on their local behaviors at each test example.
- The implementation of the LWE on the 20 Newsgroups dataset demonstrated its effectiveness and potential for future applications as it consistently gave average accuracy around 70% and an average MSE around 0.25. There is scope for improvement in the present version of the algorithm.
- Based on the work done after Mid-term Review it can be concluded that LWE algorithm failed at delivering high accuracy on the Stock Market Data that contains temporal dependencies. However, the proposed Confidence based Ensemble algorithm outperformed all other methods on the stock market data. In future the accuracy of Confidence based Ensemble algorithm can be improved by combining its prediction with a suitable algorithm instead of K means clustering.

9 References

[1] W. Fan and I. Davidson. On sample selection bias and its efficient correction via model averaging and unlabeled examples. In Proc. of SDM'07,2007, https://epubs.siam.org/doi/abs/10.1137/1.9781611972771.29

[2] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Sch¨olkopf. Correcting sample selection bias by unlabeleddata. In Proc. of NIPS' 06, pages 601–608. 2007, https://proceedings.neurips.cc/paper/2006/hash/a2186aa7c086b46ad4e8bf81e2a3a19b-Abstract.html

[3] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. Journalof Statistical Planning and Inference, 90(2):227–244, 2000, https://www.sciencedirect.com/science/article/pii/S0378375800001154

[4] Saigal, P., & Khanna, V. Multi-category news classification using Support Vector Machine based classifiers. *SN AppliedSciences*, *2*(3), 458., (2020), https://link.springer.com/article/10.1007/s42452-020-2266-6

[5] Dai, W., Xue, G. R., Yang, Q., & Yu, Y. (2007, August). Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 210- 219).,2007, https://dl.acm.org/doi/abs/10.1145/1281192.1281218

[6] Ieracitano, C., Adeel, A., Gogate, M., Dashtipour, K., Morabito, F. C., Larijani, H., ... & Hussain, A. (2018). Statistical analysis driven optimized deep learning system for intrusion detection. In *Advances in Brain Inspired Cognitive Systems: 9th International Conference, BICS 2018, Xi'an, China, July 7-8, 2018, Proceedings 9* (pp. 759-769). SpringerInternational Publishing.,2018, https://researchonline.gcu.ac.uk/ws/portalfiles/portal/26504106/BICS_2018_paper_65.pdf

[7] Danny Vanpoucke, https://github.com/DannyVanpoucke/LSSVMlib/blob/master/LSSVMlib/LSSVMRegression.py,Accessed on 20th March, 2024.